REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
17-11-2009	Final Performance Report	15-07-2006 to 31-07-2009

4. TITLE AND SUBTITLE
Acquisition and Use of Internal Models for Human Motor Learning

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-06-1-0492

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Robert Jacobs

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Rochester
517 Hylan Building
Rochester, NY 14627-0140

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
Arlington, VA

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-OSR-VA-TR-2012-0553

12. DISTRIBUTION/AVAILABILITY STATEMENT
A-Approve for public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We study decision making in dynamic environments in general, and human motor learning in particular. Our approach focuses on the acquisition and use of libraries of representational primitives. This approach is motivated by computational considerations -- learning new motor plans by linearly combining primitives from a library ameliorates the "curse of dimensionality". It is also motivated by evidence from the field of cognitive neuroscience indicating that biological organisms (including humans) linearly combine motor primitives (known as motor synergies) when planning and executing motor actions. We have made excellent progress showing that linear combinations of "global" primitives can achieve near-optimal performance on tasks requiring the control of a simulated two-joint robot arm. We have also shown that new linear combinations for novel tasks can be learned rapidly. In more recent research, we have explored the strengths of libraries of "local" primitives where primitives are linearly combined using a "local" additive regression procedure.

15. SUBJECT TERMS
motor control; motor primitives

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Robert Jacobs
					19b. TELEPHONE NUMBER *(Include area code)* 585-275-0753

LETTER Communicated by Andrea D'Avella

Properties of Synergies Arising from a Theory of Optimal Motor Behavior

Manu Chhabra
Department of Computer Science, University of Rochester, Rochester, NY 14627, U.S.A.

Robert A. Jacobs
Department of Brain and Cognitive Sciences, University of Rochester, Rochester, NY 14627, U.S.A.

We consider the properties of motor components, also known as synergies, arising from a computational theory (in the sense of Marr, 1982) of optimal motor behavior. An actor's goals were formalized as cost functions, and the optimal control signals minimizing the cost functions were calculated. Optimal synergies were derived from these optimal control signals using a variant of nonnegative matrix factorization. This was done using two different simulated two-joint arms—an arm controlled directly by torques applied at the joints and an arm in which forces were applied by muscles—and two types of motor tasks—reaching tasks and via-point tasks.

Studies of the motor synergies reveal several interesting findings. First, optimal motor actions can be generated by summing a small number of scaled and time-shifted motor synergies, indicating that optimal movements can be planned in a low-dimensional space by using optimal motor synergies as motor primitives or building blocks. Second, some optimal synergies are task independent—they arise regardless of the task context—whereas other synergies are task dependent—they arise in the context of one task but not in the contexts of other tasks. Biological organisms use a combination of task-independent and task-dependent synergies. Our work suggests that this may be an efficient combination for generating optimal motor actions from motor primitives. Third, optimal motor actions can be rapidly acquired by learning new linear combinations of optimal motor synergies. This result provides further evidence that optimal motor synergies are useful motor primitives. Fourth, synergies with similar properties arise regardless if one uses an arm controlled by torques applied at the joints or an arm controlled by muscles, suggesting that synergies, when considered in "movement space," are more a reflection of task goals and constraints than of fine details of the underlying hardware.

1 Introduction

Marr (1982) defined three levels of analysis of a complex information processing device. The top level, known as the computational theory, examines what the device does and why. A distinguishing feature of this level is that it provides an explanation for why a device does what it does by studying the device's goals. Although there may be many different ways of developing a computational theory of aspects of human behavior, an increasingly popular way is through optimal models that formalize goals as mathematical constraints or criteria, search for behaviors that optimize the criteria, and compare the optimal behaviors with human behaviors. If there is a close match, then it is hypothesized that people are behaving as they do because they are efficiently satisfying the same goals as were built into the optimal model.

This approach is commonplace in the study of human motor behavior (see Todorov, 2004, for a review). Flash and Hogan (1985), for example, proposed an optimal model of how people plan trajectories for reaching movements. This model emphasizes that trajectories should be smooth—the model searches for trajectories that minimize the jerk of a movement (i.e., the third derivative of position with respect to time). It is able to explain the fact that reaches tend to move along straight lines and tend to have bell-shaped velocity profiles. Harris and Wolpert (1998) developed an optimal model of motor control that attempts to minimize the variance of the point reached at the end of a movement despite motor noise whose magnitude is dependent on the size of the control signals. They showed that this model explains several aspects of both eye movements and hand reaches.

This article is concerned with motor synergies arising from a computational theory (in the sense of Marr, 1982) of optimal motor behavior. To understand motor synergies, it is helpful to first understand the degrees of freedom problem (Bernstein, 1967). Biological motor systems typically have many degrees of freedom, where the degrees of freedom in a system are the number of dimensions in which the system can independently vary (Rosenbaum, 1991). Because the number of degrees of freedom of a system carrying out a task often exceeds the number of degrees of freedom needed to specify the task, the degrees of freedom are typically redundant (Jordan & Rosenbaum, 1989). Consider, for example, the problem of touching the tip of your nose. The location of your nose has three degrees of freedom (its x, y, and z position in Cartesian coordinates), but the joints of your arm have seven degrees of freedom (the shoulder has three degrees of freedom, and the elbow and wrist each have two). Consequently, there are many different settings of your arm's joint positions that all allow you to touch your nose. Which setting should you use?

A solution to this problem is to create motor synergies, which are dependencies among dimensions of the motor system. For example, a motor synergy might be a coupling of the motions of your shoulder and elbow.

Motor synergies provide two types of benefits to motor systems. First, synergies ameliorate the problem of redundancy—they can constrain the set of possible shoulder, elbow, and wrist positions that allow you to touch your nose. Second, synergies reduce the number of degrees of freedom that must be independently controlled, thereby making it easier to control a motor system (Bernstein, 1967). Because synergies make motor systems easier to control, they are often hypothesized to serve as motor primitives, building blocks, or basis functions: they provide fundamental units of motor behavior that can be linearly combined to form more complex units of behavior.

Investigators of motor control are attempting to develop a comprehensive understanding of biological motor synergies. Typically, these researchers analyze neuroscientific or behavioral data using mathematical techniques in order to derive the motor synergies used by an organism. Sanger (1995) analyzed people's cursive handwriting using principal component analysis (PCA) to discover their motor synergies. He showed that linear combinations of these synergies closely reconstructed human handwriting. Thoroughman and Shadmehr (2000) studied people's motor learning behaviors to derive motor synergies based on gaussian radial basis functions. They showed that linear combinations of these synergies matched people's behaviors when adapting to new environmental conditions. Mussa-Ivaldi, Giszter, and Bizzi (1994) identified frogs' motor synergies by stimulating sites in their spinal cords and verified that stimulation of two sites leads to the vector summation of the forces generated by stimulating each site separately.

A possible confusion in the motor control literature is that synergies derived from neuroscientific or behavioral data using mathematical techniques are sometimes referred to as "optimal." For example, Sanger (1995) derived synergies from human behavioral data using PCA, a linear optimal dimensionality-reduction technique, and referred to the results as "optimal movement primitives." It is important to keep in mind, however, that these synergies arise from optimal analysis of people's actions and are not necessarily the same ones as would arise from a computational theory (again, in the sense of Marr, 1982) of optimal motor behavior. Based on the discussion above, a computational theory might involve a model that formalizes the actor's goals as mathematical criteria and searches for the actions that optimize the criteria. An optimal analysis of the optimal actions could then derive the motor synergies. Synergies discovered in this way would be "optimal" in the sense that they arise from a computational theory of optimal motor behavior.

To date, we know of only one study of motor synergies that arise from a computational theory. Todorov and Jordan (2002) proposed a computational theory that uses an optimal feedback controller as a model of motor coordination and noted that this controller produces motor synergies. In brief, the controller implements the "principle of minimal intervention"—it

does not attempt to control a system along dimensions that are irrelevant for a task. Because the system's degrees of freedom are controlled along some task dimensions but not others, couplings or synergies among the degrees of freedom arise. Todorov and Jordan thereby explained the emergence of synergies.

Although Todorov and Jordan (2002) explained the emergence of synergies; they did not study the specific properties of these synergies. In contrast, this article details the properties of synergies arising from a theory of optimal motor behavior. We have created an optimal controller for a nonlinear system that formalizes goals as mathematical constraints and searches for control signals that optimize the constraints. This was done using two different simulated two-joint arms—an arm controlled directly by torques applied at the joints and an arm in which forces are applied by muscles—and two types of motor tasks—reaching tasks (move an end effector from one point to another) and via-point tasks (move from one point to another while passing through an intermediate point). In all cases, we derived synergies from the optimal control signals using an extension to nonnegative matrix factorization (d'Avella, Saltiel, & Bizzi, 2003) and studied the properties of these synergies.

Our studies of the resulting motor synergies reveal several interesting findings. First, optimal motor actions can be generated by summing a small number of scaled and time-shifted motor synergies, indicating that optimal movements can be planned in a low-dimensional space by using optimal motor synergies as motor primitives or building blocks. Second, some optimal synergies are task independent—they arise regardless of the task context—whereas other synergies are task dependent—they arise in the context of one task but not in the contexts of other tasks. Biological organisms use a combination of task-independent and task-dependent synergies. Our work suggests that this may be an efficient combination for generating optimal motor actions from motor primitives. Third, optimal motor actions can be rapidly acquired by learning new linear combinations of optimal motor synergies. This result provides further evidence that optimal motor synergies are useful motor primitives. Fourth, synergies with similar properties arise regardless if one uses an arm controlled directly by torques applied at the joints or an arm controlled by muscles, suggesting that synergies, when considered in "movement space," are more a reflection of task goals and constraints than of fine details of the underlying hardware.

2 Computing the Optimal Control Signals

We simulated a two-joint arm that can be characterized as a second-order nonlinear dynamical system (e.g., Hollerbach & Flash, 1982):

$$\mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta}) + \mathcal{B}\dot{\theta} = \tau \tag{2.1}$$

Table 1: Parameter Values Used in the Simulation of a Two-joint Arm.

Parameter	Value	Parameter	Value
b_{11}	0.05 kgm^2 s^{-1}	b_{22}	0.05 kgm^2 s^{-1}
b_{21}	0.025 kgm^2 s^{-1}	b_{12}	0.025 kgm^2 s^{-1}
m_1	1.4 kg	m_2	1.0 kg
l_1	0.30 m	l_2	0.33 m
I_1	0.025 kgm^2	I_2	0.045 kgm^2
s_1	0.11 m	s_2	0.16 m

Note: The parameters $\{b_{ij}\}$ denote the (i, j)th elements of a joint friction matrix; m_i, l_i, and I_i denote the mass, length, and moment of inertia of the ith link, respectively; and s_i denotes the distance from the ith joint to the ith link's center of mass.

where τ is a vector of torques, θ is a vector of joint angles, $\mathcal{M}(\theta)$ is an inertial matrix, $\mathcal{C}(\theta, \dot{\theta})$ is a vector of coriolis forces, and $\mathcal{B}$ is a joint friction matrix. We used the same parameter values for the arm as Li and Todorov (2004). These values are listed in Table 1.

We studied two types of tasks: reaching tasks and via-point tasks. In a reaching task, the arm must be controlled so that its end effector moves from a start location to a target location. A via-point task is identical except that there is an additional requirement that the end effector also move through an intermediate location known as a via-point.

For any reaching or via-point task, there are many time-varying torque vectors $\tau(t)$ that will move the arm so that it successfully performs the task. As discussed above, this multiplicity of control solutions is due to redundancy in the two-joint arm and is known as the degrees-of-freedom problem. How do we choose a particular solution? According to the optimality framework, an actor's goals are formalized as mathematical constraints that are combined in a cost function, and an optimal control signal is a signal that minimizes this function.

For the reaching task, we used the following cost function,

$$J(\tau(t)) = \frac{1}{2} \parallel e(T) - e^* \parallel^2 + k_1 \parallel \dot{e}(T) \parallel^2 + \frac{k_2}{2} \int_0^T \tau(t)^T \tau(t) dt, \qquad (2.2)$$

where k_1 and k_2 are constants (we used the same values as Todorov & Li, 2005: $k_1 = 0.001$ and $k_2 = 0.0001$), T is the duration of the movement, $e(T)$ is the end-effector location at time T, and e^* is the target location at time T. The first term penalizes reaches that deviate from the target location, the second term penalizes reaches that do not have a zero velocity at the end of the movement, and the third term penalizes reaches that require large torques (or "energy"). This cost function has previously been used by Li and Todorov (2004; see also Todorov & Li, 2005). Minimization of this function results in control signals that produce reaches with several properties of

natural movements, including bell-shaped velocity profiles, lower velocities at higher curvatures, and near-zero velocities at the beginnings and ends of movements.

For the via-point task, we modified the above cost function to also penalize movements that do not pass through the via-point midway through the movement. The cost function has the form

$$J(\tau(t)) = \frac{1}{2} \| e(T) - e^* \|^2 + \frac{1}{2} \| e(T/2) - e_v^* \|^2 + k_1 \| \dot{e}(T) \|^2 + \frac{k_2}{2} \int_0^T \tau(t)^T \tau(t) dt, \quad (2.3)$$

where e_v^* is the via-point or desired end-effector location at the middle of the movement. This function penalizes reaches that deviate from the via-point at time $T/2$.

To find the optimal control signal for a reaching or via-point task, the corresponding cost function must be minimized. Unfortunately, when using nonlinear systems such as the two-joint arm described above, this minimization is computationally intractable. Researchers typically resort to approximate methods to find locally optimal solutions. We used one such method, known as the iterative linear quadratic regulator (iLQR), developed by Li and Todorov (2004; see also Todorov & Li, 2005). We now briefly summarize this method in a generic setting.

A continuous-time linear dynamical system is given by

$$\dot{x}(t) = f(x(t), u(t)), \quad (2.4)$$

where x is the state of the system and u is the input control signal. For the two-joint arm described above, the state x is $(\theta, \dot{\theta})^T$, and the control u is τ. Consider a cost function of the form

$$J(u(t)) = \sum_i h_i(x(t_i)) + \int_0^T l(x(t), u(t)) dt. \quad (2.5)$$

Note that the cost functions for the reaching and via-point tasks are of this form. In the cost function for the reaching task, for example, the two discrete penalties (deviation of the end-effector location from the target location at time T and deviation of the end-effector velocity from zero at time T) correspond to the first term on the right-hand side of equation 2.5, and a continuous energy-like cost for large torques corresponds to the second term.

The iLQR starts with an initial guess of the optimal control signal and iteratively improves it. From the control signal $u_i(t)$ at iteration i, the trajectory $x_i(t)$ is computed using a standard Euler approximation. The algorithm

uses three steps to find the control signal for the next iteration. It starts by linearly approximating the dynamical system given in equation 2.4 and quadratically approximating the cost function given in equation 2.5. These approximations are made around $(x_i(t), u_i(t))$ at each time step t. Using these approximations, a modified linear quadratic gaussian (LQG) control problem is then formulated in the $(\delta x, \delta u)$ space, where $x + \delta x$ and $u + \delta u$ are improved approximations to x and u, respectively. This formulation is valid only where these approximations are accurate: a small region around $(x_i(t), u_i(t))$. Finally, the optimal correction to control $u_i(t)$ at iteration i, denoted $\delta u_i^*(t)$, is computed by solving this modified LQG problem. This step requires the solution to a modified Riccatti-like set of equations. Fortunately, finding this solution is computationally efficient. Once the optimal corrections have been obtained, the algorithm sets $u_{i+1}(t) = u_i(t) + \delta u_i^*(t)$ and proceeds to the next iteration. The algorithm stops if there is no significant improvement in the trajectory. The end result is a locally optimal trajectory x^* and locally optimal control signal u^*.

We have found that the iLQR works well on both reaching and via-point tasks when using the two-joint arm. Figure 1 shows examples of optimal reaching (top row) and via-point (bottom row) movements computed by the iLQR. The graphs in the left column show the movement of the end effector (horizontal and vertical axes give the x and y coordinates of the end effector in Cartesian space), whereas the graphs in the right column show the velocity profiles (horizontal axes represent time, and vertical axes represent velocity of the end effector). Clearly, the iLQR produces smooth movements with bell-shaped velocity profiles. In addition, the velocity profile for the via-point movement (bottom-right graph) indicates that end-effector velocity decreases with increasing path curvature.

3 Obtaining Optimal Synergies

As discussed above, motor synergies are dependencies among dimensions of a motor system. They are useful because they can ameliorate the problem of redundancy and because they reduce the number of degrees of freedom that must be independently controlled, thereby making it easier to control a motor system. Synergies are often hypothesized to serve as motor primitives, building blocks, or basis functions.

Researchers have used a variety of methods to compute motor synergies. We used a variant of nonnegative matrix factorization developed by d'Avella et al. (2003). This algorithm requires two inputs. One input is the number of synergies, denoted N. The other is a matrix of control signals, where each control signal is a $2 \times T$ matrix of optimal torques computed by the iLQR for a given task (this matrix has $2 \times T$ elements because torques are applied to both joints of the two-joint arm at each time step of a movement, and there are T time steps per movement). The input matrix of control signals is a vertical stack of individual control signal matrices.

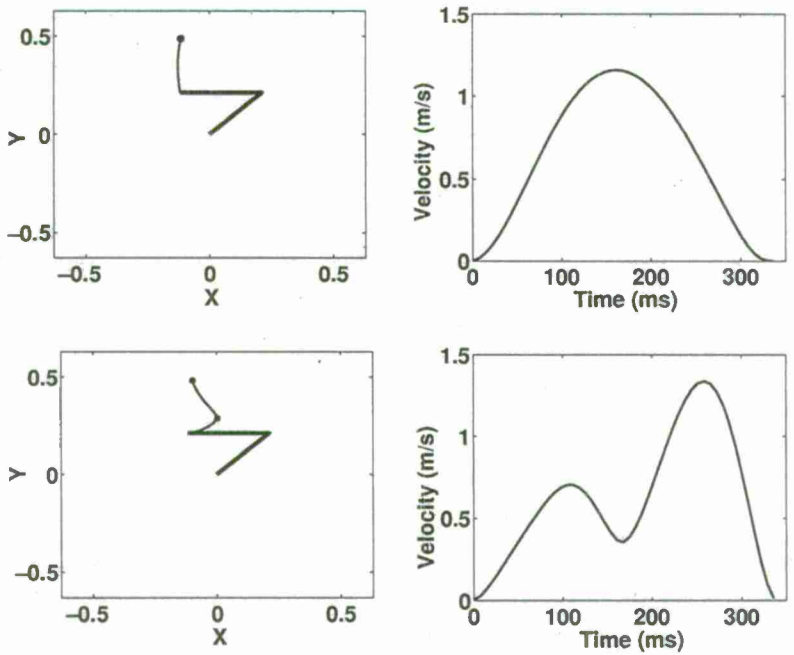

Figure 1: Examples of optimal reaching (top row) and via-point movements (bottom row) computed using the iLQR. The graphs in the left column show the movement of the end effector: the horizontal and vertical axes represent the x and y coordinates of the end effector in Cartesian space. The thick lines show the orientation of the two-joint arm at the start of the movement, and the thin lines show the path of the end effector. The graphs in the right column show the velocity profiles: the horizontal axes represent time, and the vertical axes represent velocity of the end effector.

For example, if the iLQR was used to find the optimal control signals for 500 reaching tasks (tasks with different initial configurations of the arm or different target locations) and each reach had a duration of 400 time steps, then the matrix would consist of 1000 rows where each block of two rows is a 2×400 element matrix giving the optimal torques for each joint at each time step of a reach. As its output, the algorithm seeks a set of synergies such that every control signal can be expressed as a sum of scaled and time-shifted synergies. Mathematically, it seeks a set of N synergies, denoted $\{\mathbf{w}_i, i = 1, \ldots, N\}$, such that control signal $\mathbf{m}$ can be written as follows,

$$\mathbf{m}(t) = \sum_{i=1}^{N} c_i \mathbf{w}_i (t - t_i), \tag{3.1}$$

where $\{c_i, i = 1, \ldots, N\}$ is a set of coefficients that scale the synergies, and $\{t_i, i = 1, \ldots, N\}$ is a set of times that time-shift the synergies. The algorithm searches for the synergies, scaling coefficients, and time shifts that minimize the sum of squared errors between the actual control signals and the reconstructed signals.

A technical detail is that the algorithm requires a set of nonnegative control signals (each element of a control vector must be nonnegative). In our case, a torque vector might have negative elements. We overcame this problem in a manner inspired by biological motor systems' use of agonist and antagonist muscles to apply torques at joints. We recoded a 2×1 torque vector as a 4×1 vector in which the first two elements give the anticlockwise and clockwise torques for the first joint (shoulder), and the last two elements provide the same information for the second joint (elbow).[1] For example, if torque $(2, -1)^T$ is applied to the joints, it means that a $+2$ torque is applied to the first joint in the anticlockwise direction, and a $+1$ torque is applied to the second joint in the clockwise direction. We recoded this torque vector to the nonnegative vector $(2, 0, 0, 1)^T$.

4 Simulation Results

This section reports the results of seven experiments. The first four experiments used the two-joint arm described above in which torques were applied at the joints. The last three experiments used the same arm, except forces were applied by muscles.

All experiments used the same collection of reaching and via-point tasks. We created 320 instances of each task as follows. Ten initial positions of the arm were randomly generated by uniformly sampling the first joint angle from the set $[-\pi/4, \pi/2]$ and the second joint angle from the set $[0, 3\pi/4]$. For each initial position, 32 target locations were generated. A target was generated by randomly selecting a movement distance (sampled uniformly from the range 10–50 cm) and an angle of movement (sampled uniformly from the range 0–2π). For the via-point task, a via-point was placed at a random angle (sampled uniformly from the set $[-\pi/3, \pi/3]$) from the line joining the initial and target locations. The via-point's distance from the

[1] Below we present results in which we consider the number of motor synergies required to reconstruct optimal movements with small error when an arm is controlled by torques applied directly to its joints. It is possible that the operation of mapping a two-dimensional vector with real values to a four-dimensional vector with nonnegative values introduces a bias into the estimate of this number. Nonetheless, our use of the mapping is justified as follows. We wish to compare synergies obtained when an arm is controlled by torques applied directly to its joints with synergies obtained when an arm is controlled by forces applied by muscles. Therefore, it is necessary to use the same representational format and dimensionality-reduction algorithm for obtaining synergies in both cases. When an arm is controlled by muscles, synergies are extracted on the basis of muscle activations, which must be nonnegative values.

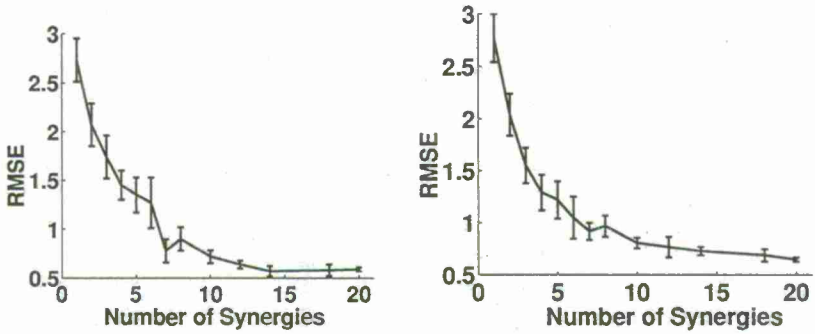

Figure 2: The graphs plot the root mean squared error (RSME) between actual and reconstructed test items for reaching (left graph) and via-point (right graph) tasks as a function of the number of synergies used in the reconstructions. The error bars give the standard errors of the means.

initial location was selected randomly to be between one-third and two-thirds of the distance between initial and target locations. The duration of a movement was 350 msec, and new torques were applied every 7 msec.

4.1 Experiment 1: A Small Set of Synergies Can Reconstruct Optimal Movements. The first experiment evaluated whether optimal reaching or via-point control signals can be expressed as a sum of a small number of scaled and time-shifted synergies. If so, then the synergies can be regarded as useful motor primitives.

For each type of task, the iLQR was applied to each instance of the task to generate 320 optimal control signals. These signals were divided into five equal-sized sets, which were then used by a fivefold cross-validation procedure to create training and test data items. Four sets of control signals were used for training, and the remaining set was used for testing. This was repeated for all five such combinations of training and test sets. During training, nonnegative matrix factorization was used as described above to discover a set of synergies. During testing, these synergies were time-shifted and linearly combined to reconstruct the test control signals. Nonnegative matrix factorization was used to find the time shifts and linear coefficients.

The results for the reaching and via-point tasks are shown in the left and right graphs of Figure 2, respectively. The horizontal axes give the number of synergies. The vertical axes give the root mean squared error (RMSE) between actual and reconstructed test control signals. The error bars show the standard errors of the means. With both reaching and via-point tasks, the error is near its minimum when relatively few synergies (about six or seven) were used. For our purposes, this is an important result

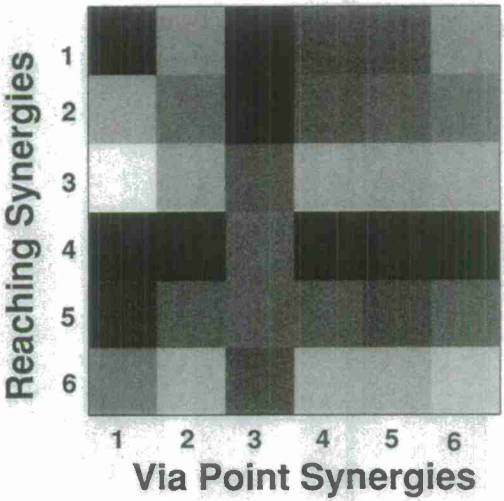

Figure 3: The similarity matrix when six synergies were obtained from the reaching task and the via-point task. The lightness of the square at row i and column j gives the cosine of the angle between the ith reaching-task synergy vector and the jth via-point task synergy vector: white is a value of 1, black is a value of 0, and intermediate gray-scale values represent intermediate values.

because it means that the synergies are useful motor primitives: optimal movements can be planned in a relatively low-dimensional space by time-shifting and linearly combining a small number of synergies. Furthermore, the fact that the error curves for the reaching and via-point tasks are similar suggests that these tasks have similar task complexity. This is surprising because generating optimal via-point movements intuitively seems more complicated than generating optimal reaches.

4.2 Experiment 2: Task-Independent and Task-Dependent Synergies. The second experiment evaluated whether optimal motor synergies are task independent or task dependent. This issue is interesting due to recent neurophysiological findings. d'Avella and Bizzi (2005), for example, recorded electromyographic activity from 13 muscles of the hind limbs of frogs performing jumping, swimming, and walking movements. An analysis of the underlying motor synergies revealed that some synergies were used in all types of movements, whereas other synergies were movement dependent.

Figure 3 shows the similarity matrix when six synergies were obtained for the reaching task and six synergies were obtained for the via-point task. The lightness of the square at row i and column j gives the cosine of the angle between the ith reaching-task synergy vector and the jth via-point

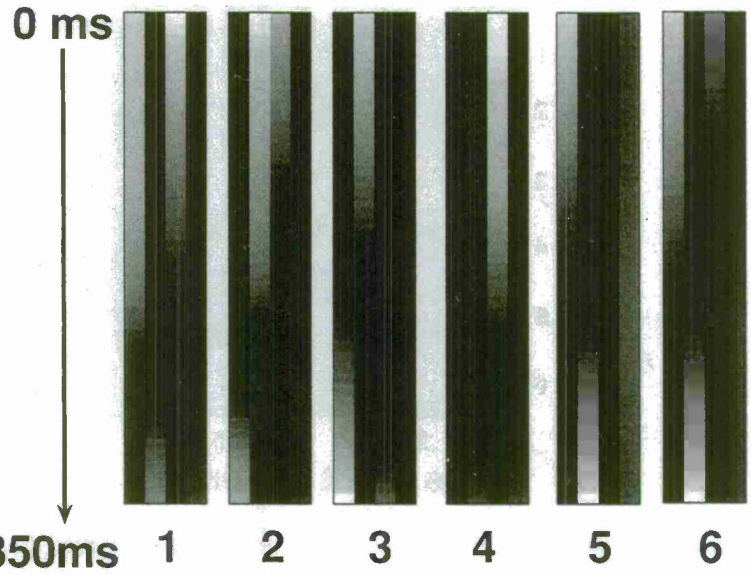

Figure 4: The six synergies obtained for the reaching task. Each synergy is represented by four columns; the first two columns represent the anticlockwise and clockwise torques for the first joint (shoulder), whereas the second two columns represent this same information for the second joint (elbow). Torques were linearly scaled to the interval [0, 1]. White indicates a torque of 1, black indicates a torque of 0, and intermediate shades of gray represent intermediate values.

task synergy vector: white is a value of 1, black is a value of 0, and intermediate gray-scale values represent intermediate values. Some synergies, such as the third reaching-task synergy and the first via-point task synergy, are highly similar, indicating that these synergies are task independent. In contrast, other synergies, such as the fourth reaching-task synergy or the third via-point task synergy, are dissimilar from all other synergies, indicating that they are task dependent. This result suggests that the combination of task-independent and task-dependent synergies found in biological organisms (e.g., d'Avella & Bizzi, 2005; Jing, Cropper, Hurwitz, & Weiss, 2004) may be efficient for generating optimal motor actions from motor synergies.

4.3 Experiment 3: Visualizing Synergies. In experiment 3, we obtained synergies for the purpose of visualizing the movements induced by these synergies. Using our collections of instances of each type of task, six synergies for the reaching task and six synergies for the via-point task were calculated as described above. The scaling coefficients for the reaching-task

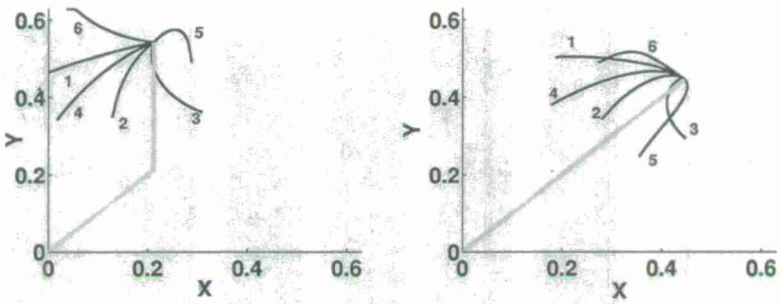

Figure 5: Movements induced by six synergies obtained for the reaching task. The horizontal and vertical axes of each graph give the x and y coordinates of the end effector in Cartesian space, the gray lines show the initial configuration of the arm, the black lines show the movements of the end effector, and the number next to each movement indicates the synergy that was applied (using the same labels as Figure 4). The left and right graphs illustrate induced movements when the initial configuration of the arm was near the center of the work space or at a far edge of the work space, respectively.

synergies or the via-point task synergies were set to their average values over the collection of reaching tasks or via-point tasks, respectively. The time-shift parameters were set to zero.

The six synergies obtained for the reaching task are illustrated in Figure 4. The horizontal axis labels the synergies, and the vertical axis depicts time. Each synergy is represented by four columns; the first two columns represent the anticlockwise and clockwise torques for the first joint, whereas the second two columns represent this same information for the second joint. Torques were linearly scaled to the interval [0, 1]. White indicates a torque of 1, black indicates a torque of 0, and intermediate shades of gray represent intermediate values.

Figure 5 illustrates movements based on these synergies. The left graph shows the induced movements when the initial arm configuration was near the center of the workspace. The horizontal and vertical axes of the graph give the x and y coordinates of the end-effector in Cartesian space, the gray lines show the initial configuration of the arm, the black lines show the movements of the end effector, and the number next to each movement indicates the synergy that was applied (using the same labels as Figure 4). The induced movements tend to be relatively straight (though some are curved) and tend to cover a wide range of directions. The right graph of Figure 5 shows the induced movements when the initial arm configuration was at a far edge of the work space. Again, the movements tend to be relatively straight. As should be expected, movements in this case are directed toward the center of the work space. Figure 5 demonstrates that synergies tend to broadly cover all possible directions of motion.

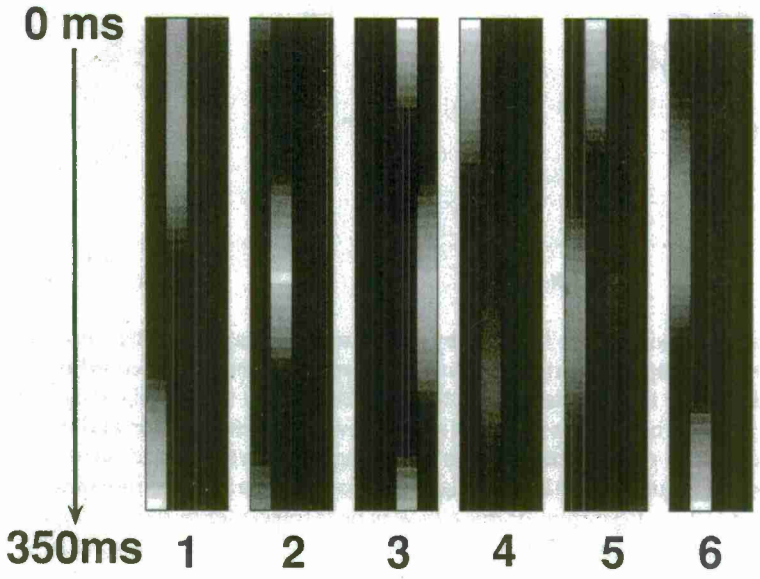

Figure 6: The six synergies obtained for the via-point task.

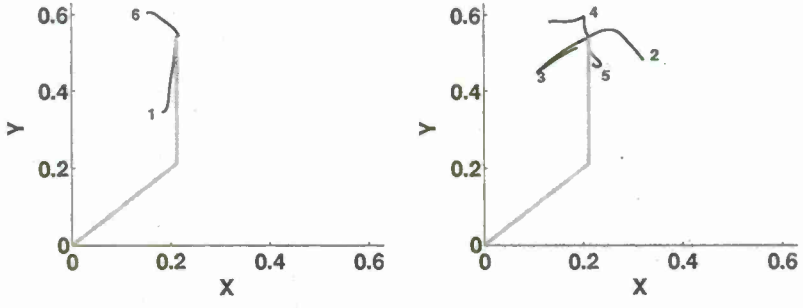

Figure 7: Movements induced by synergies obtained from the via-point task. The left and right graphs illustrate movements induced by task-independent and task-dependent synergies, respectively. The number next to each movement indicates the synergy that was applied (using the same labels as Figure 6).

The six synergies obtained for the via-point task are illustrated in Figure 6. It uses the same format as Figure 4. Figure 7 illustrates movements based on these synergies. The left graph illustrates movements induced by two synergies that were highly similar to synergies obtained from the reaching task—that is, these are task-independent synergies. The induced movements are relatively straight. Consequently, the underlying

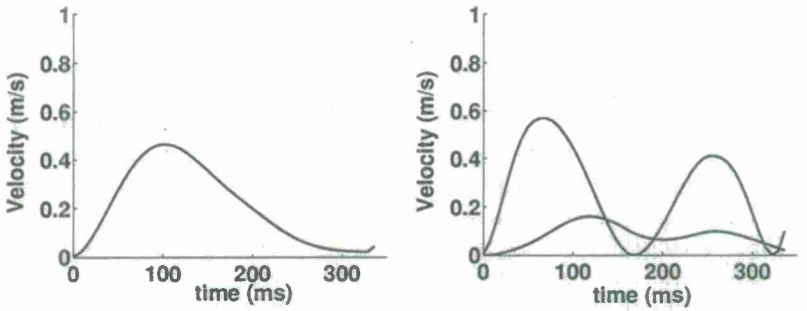

Figure 8: Velocity curves for induced movements. The left graph plots the velocity curve for a movement based on a synergy obtained from the reaching task. The right graph plots the velocity curves for two movements based on two task-dependent synergies obtained from the via-point task.

synergies are useful for both reaching and via-point tasks. The right graph illustrates movements based on four synergies that are task dependent; these synergies were not similar to synergies obtained from the reaching task. The induced movements tend to be almost piecewise linear, with a region of large curvature near the middle of the movement that is preceded and followed by regions of relatively straight motion.

Figure 8 shows the velocity curves (velocity at each moment in time) for induced movements. The left graph plots the velocity curve for a movement based on a synergy obtained from the reaching task. This curve has a bell-shaped profile, which is commonly found for reaching movements. The right graph plots the velocity curves for two movements based on two task-dependent synergies obtained from the via-point task. The shapes of these curves are typical for via-point movements.

In summary, we find that the synergies for reaching and via-point movements have intuitive forms. Movements based on synergies obtained from the reaching task tend to be straight, to broadly cover the directions available to the arm based on its initial configuration, and to have bell-shaped velocity profiles. Movements based on task-independent via-point synergies tend to have these same properties. In contrast, movements based on task-dependent via-point synergies tend to have a piecewise-linear shape with a region of high curvature near the middle of the movement and have velocity profiles with two bell shapes.

4.4 Experiment 4: Learning with Synergies. Experiment 4 evaluated whether the use of optimal motor synergies makes it easier to learn to perform new optimal motor actions. If motor synergies are useful motor primitives, then this ought to be the case.

The task was to learn to generate a reaching movement starting from an initial configuration of the arm so that the arm's end point reached a randomly selected target location. When synergies were used, control signals were expressed as linear combinations of synergies (to minimize computational demands, we did not time-shift synergies), meaning that the parameter values that needed to be learned were the linear coefficients. When synergies were not used, the values that needed to be learned were the torques applied to each joint at each moment in time.

From a collection of 320 instances of the reaching task, fivefold cross validation was used to create training and test sets. Policy gradient, a type of reinforcement learning algorithm, was used to learn estimates of the relevant parameter values (Sutton, McAllester, Singh, & Mansour, 2000). This algorithm was applied for 300 iterations. Learning with synergies occurred as follows. We calculated the optimal movements for each instance in a training set using the iLQR, and obtained four motor synergies using nonnegative matrix factorization. The policy gradient algorithm was then used to learn to perform each instance of the reaching task in the test set. At each iteration of the learning process, we numerically computed the derivatives of the reaching-task cost function (see equation 2.2) with respect to the linear coefficients used in the linear combination of synergies and performed gradient descent with the constraint that the coefficients had to be nonnegative. When learning without synergies, we computed the derivatives of the reaching-task cost function with respect to the torques at each joint and at each time step and performed gradient descent. Step sizes or learning rates that produced near-optimal performance were used when performing gradient descent with and without synergies.

The results for a typical instance of a reaching task from a test set are shown in Figure 9. The graph on the left shows the learning curves for learning with and without motor synergies. The horizontal axis gives the iteration number, and the vertical axis gives the value of the reaching-task cost function. Whereas learning without synergies was slow and never achieved good performance, learning with synergies was rapid and achieved excellent performance. Indeed, learning with synergies achieved roughly the same cost as the iLQR. The graph on the right shows the movements learned with and without synergies in Cartesian coordinates and the movement calculated by the iLQR. The movement learned without synergies never reached the target location, whereas the movement learned with synergies did. Overall, the results indicate that optimal synergies are useful motor primitives or building blocks in the sense that their use in linear combinations leads to rapid and accurate acquisition of new optimal motor actions.

4.5 Experiment 5: Motor Synergies When Forces Are Applied by Muscles. Whereas experiments 1 to 4 simulated a two-joint arm controlled directly by torques applied at the joints, experiments 5 to 7 simulated the

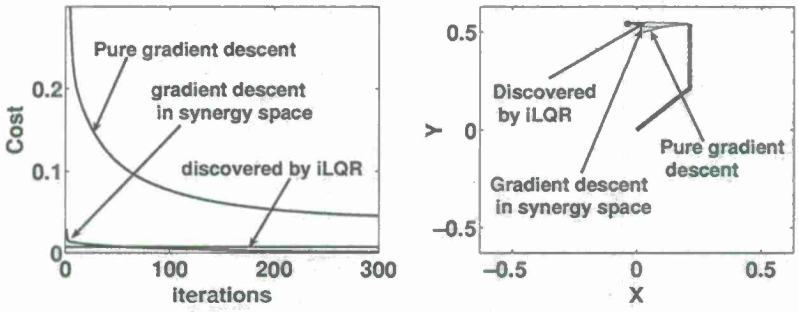

Figure 9: The graph on the left shows the learning curves for learning with and without motor synergies on a typical instance of a reaching task. The horizontal axis gives the iteration number, and the vertical axis gives the value of the reaching-task cost function. The graph on the right shows the movements learned with and without synergies in Cartesian space and the movement calculated by the iLQR.

same arm except forces were applied by muscles. We conducted experiments with muscles so that we could verify that the results reported above are also valid when simulating more complex and biologically realistic systems such as an arm controlled by muscles.

We used the muscle model developed by Todorov and Li (2005; see also Brown, Cheng, & Leob, 1999). In brief, this model uses six muscles that apply forces to a two-joint arm. The control signal is the neural input to the muscles. This input passes through a nonlinear low-pass filter to produce muscle activations. The tension of a muscle is a function of the muscle's current activation, length, and length velocity. The tension produces forces on the arm's links that, in turn, produce joint torques. Note that this system is significantly more complicated than the arm in which torques are applied directly at the joints. This system has a six-dimensional control space (neural input to each of six muscles), a 10-dimensional state space (six muscle activations and the angular position and velocity of each joint), muscle activations that might saturate, and dynamics with temporal delays (due to the low-pass filtering of neural input).

In experiments 1 to 4, nonnegative matrix factorization was applied to the optimal control signals to obtain motor synergies. In contrast, this factorization was not applied to the control signals—the neural input—in experiments 5 to 7; rather, it was applied to the optimal muscle activations. We found that the factorization procedure was significantly more robust when applied to the muscle activations due to the smoothness of their values (recall that the activations are low-pass accumulations of the neural inputs). Factorization of muscle activations was also conducted by d'Avella et al. (2003).

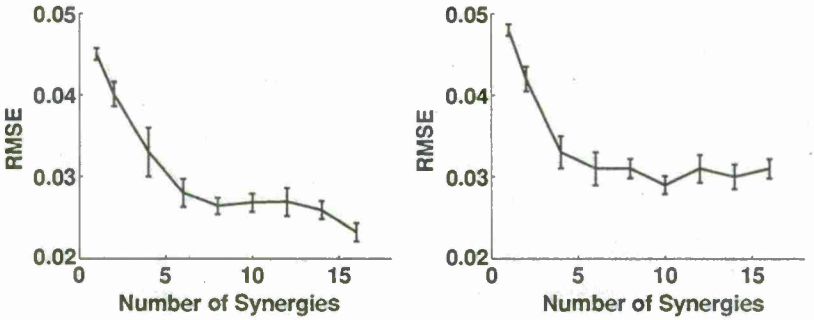

Figure 10: The graphs plot the root mean squared error (RMSE) between actual and reconstructed test items for reaching (left graph) and via-point (right graph) tasks when using the arm controlled by muscles as a function of the number of synergies used in the reconstructions. The error bars give the standard errors of the means.

Data for experiments 5 to 7 were collected in a similar manner as for experiments 1 to 4. We created 320 instances of the reaching task and the via-point task. Fivefold cross validation was used to create training and test sets of task instances. The iLQR was used to calculate the optimal sequence of neural inputs for each training instance (the cost functions for the reaching and via-point tasks given above were suitably modified by replacing the torque vector—the control input in experiments 1 to 4—with the neural input vector—the control input in experiments 5 to 7). Optimal muscle activations were created from the optimal neural inputs by low-pass filtering. Nonnegative matrix factorization was applied to the optimal muscle activations to generate optimal synergies. Based on these synergies, nonnegative matrix factorization was also used to perform task instances from test sets by computing optimal sums of scaled and time-shifted synergies.

Experiment 5 parallels experiment 1 in the sense that it evaluated whether optimal muscle activations can be expressed as a linear combination of a small number of time-shifted synergies. The results for the reaching and via-point tasks are shown in the left and right graphs of Figure 10, respectively. With both reaching and via-point tasks, the error is near its minimum when relatively few synergies were used. We conclude that synergies are useful motor primitives because optimal movements can be planned in a relatively low-dimensional space by summing a small number of scaled and time-shifted synergies.

4.6 Experiment 6: Task-Independent and Task-Dependent Synergies When Forces Are Applied by Muscles. Experiment 6 parallels experiment 2 in the sense that it evaluated whether optimal motor synergies are task independent or task dependent (d'Avella & Bizzi, 2005; Jing et al., 2004).

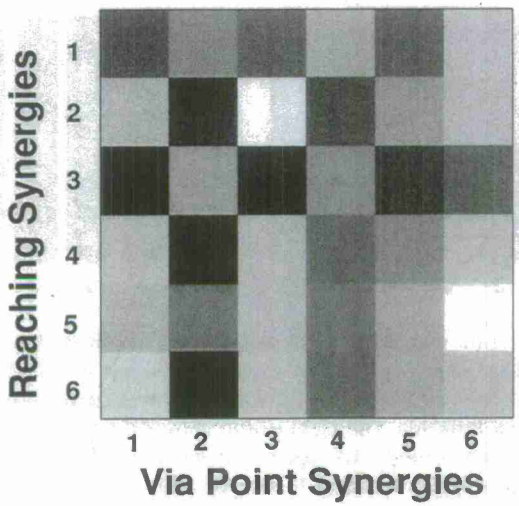

Figure 11: The similarity matrix for the six synergies obtained from the reaching task and the six synergies obtained from the via-point task using the arm controlled by muscles. The lightness of the square at row i and column j gives the cosine of the angle between the ith reaching-task synergy vector and the jth via-point task synergy vector—white is a value of 1, black is a value of 0, and intermediate gray-scale values represent intermediate values.

Figure 11 shows the similarity matrix for the six synergies obtained from the reaching task and the six synergies obtained from the via-point task. Some synergies, such as the second reaching-task synergy and third via-point task synergy are highly similar, indicating that these synergies are task independent. In contrast, other synergies, such as the third reaching-task synergy and the second via-point task synergy, are dissimilar from all other synergies, indicating that they are task dependent. A combination of task-independent and task-dependent synergies was also found in experiment 2, which used an arm controlled directly by torques. This result suggests that the combination of task-independent and task-dependent synergies found in biological organisms may be efficient for generating optimal motor actions from motor primitives.

4.7 Experiment 7: Visualizing Synergies When Forces Are Applied by Muscles. In experiment 7, we obtained synergies for the purpose of visualizing the movements induced by these synergies when forces are applied by muscles. Consequently, experiment 7 parallels experiment 3 and was conducted in an analogous manner.

Six synergies were obtained for the reaching task when the arm was controlled by forces applied by muscles. These synergies are illustrated in

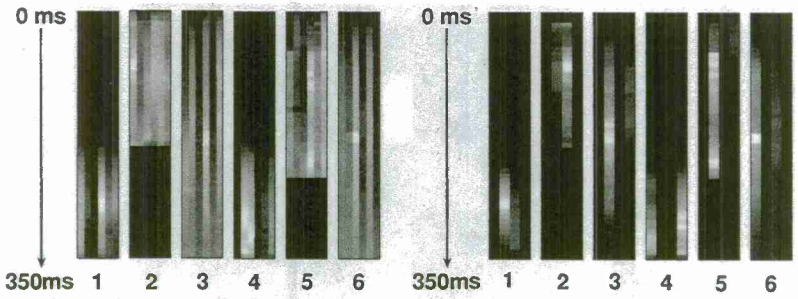

Figure 12: Six synergies were obtained for the reaching task when the arm was controlled by forces applied by muscles. The left graph shows the muscle activations. For each synergy, there are six columns corresponding to the activations of the six muscles. The right graph shows the torques generated by the synergies. For each synergy, the four columns indicate the anticlockwise and clockwise torques for joints 1 and 2, respectively.

Figure 12. The left graph shows the muscle activations. For each synergy, there are six columns corresponding to the activations of the six muscles.[2] Muscle activations were linearly scaled to the interval [0, 1]. White indicates an activation of 1, black indicates an activation of 0, and intermediate shades of gray indicate intermediate activation values. The right graph shows the torques generated by the synergies. For each synergy, the four columns indicate the anticlockwise and clockwise torques for joints 1 and 2, respectively.

Figure 13 illustrates the movements induced by these synergies. The left graph shows the induced movements when the initial arm configuration was near the center of the work space. These movements tend to be relatively straight (though some are curved) and tend to cover a wide range of directions. The right graph shows the induced movements when the initial arm configuration was at a far edge of the work space. The movements tend to be relatively straight and are directed toward the center of the work space. For our purposes, a notable feature of these induced movements is that they closely resemble the movements induced by reaching-task synergies obtained when the arm was controlled by torques applied directly at the joints (see experiment 3 above). These data are consistent with the idea that synergies, when considered in "movement space," are more a reflection of task goals and constraints than of fine details of the underlying hardware.

[2] The muscles are (1) biceps long, brachialis, brachioradialis; (2) triceps lateral, anconeus; (3) deltoid anterior, coracobrachialis; (4) deltoid posterior; (5) biceps short; and (6) triceps long. See Li and Todorov (2004) for details.

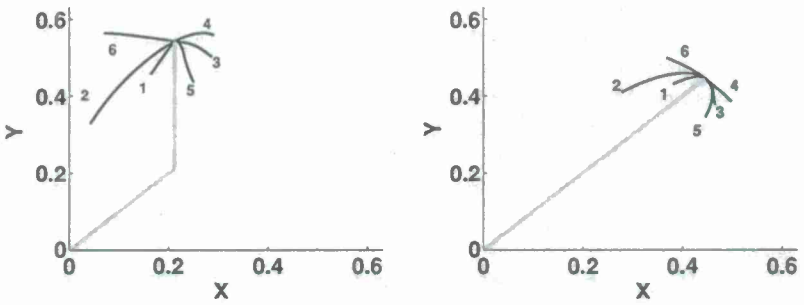

Figure 13: Movements induced by six synergies obtained for the reaching task when the arm was controlled by forces applied by muscles. The left and right graphs illustrate induced movements when the initial configuration of the arm was near the center of the work space or at a far edge of the work space, respectively. The number next to each movement indicates the synergy that was applied (using the same labels as Figure 12).

5 Discussion

In summary, this letter has considered the properties of synergies arising from a computational theory (in the sense of Marr, 1982) of optimal motor behavior. An actor's goals were formalized as cost functions, and the optimal control signals minimizing the cost functions were calculated by the iLQR. Optimal synergies were derived from these optimal control signals using a variant of nonnegative matrix factorization. This was done for both reaching and via-point tasks and for a simulated two-joint arm controlled by torques applied at the joints as well as an arm in which forces were applied by muscles. In brief, studies of the motor synergies revealed several interesting findings: (1) optimal motor actions can be generated by summing a small number of scaled and time-shifted motor synergies; (2) some optimal synergies are task independent, whereas other synergies are task dependent; (3) optimal motor actions can be rapidly acquired by learning new linear combinations of optimal motor synergies; and (4) synergies with similar properties arise regardless if one uses an arm controlled by torques applied at the joints or an arm controlled by muscles.

Future work will need to address shortcomings of our experiments. Our findings were obtained using simple motor tasks and a simple two-joint arm. We used reaching and via-point tasks because these are commonly performed movements and are frequently studied in the literature. We used a two-joint arm because it is computationally tractable. We conjecture that our basic results will still be found even with more complex tasks. This hypothesis is based on the fact that many complex movements can be regarded as combinations of simpler reaching and via-point movements.

We also conjecture that our results will still be found with more complex arms. This hypothesis is based on the fact that we obtained similar results regardless of whether we used a simple arm—a two-joint arm controlled by torques applied at the joints—or a more complex arm—a two-joint arm controlled by forces applied by muscles. Computationally, an obstacle to using more complex tasks and arms is the need to calculate optimal control signals. Using current computer technology, the calculation of optimal controls for nonlinear systems with many degrees of freedom is typically not possible.

Our findings were also obtained using specific mathematical techniques, such as the iLQR optimization method and the nonnegative matrix factorization method. We believe that our choices of mathematical techniques were reasonable. Again, this is an area in which important computational issues will need to be addressed before future studies can consider more complex motor tasks and arms. In particular, there is a need to develop improved dimensionality-reduction techniques for obtaining synergies. For example, the nonnegative matrix factorization method, like other methods, cannot be applied when movements have widely different durations and, thus, control signals have widely different dimensions. Future work will need to address this and many other unsolved problems.

Acknowledgments

We thank E. Todorov for help with the iLQR optimal control algorithm and two anonymous reviewers for helpful comments on an earlier version of this article. This work was supported by NIH research grant R01-EY13149.

References

Bernstein, N. (1967). *The coordination and regulation of movements*. London: Pergamon.

Brown, I. E., Cheng, E. J., & Leob, G. E. (1999). Measured and modeled properties of mammalian skeletal muscle; II: The effects of stimulus frequency on forcelength and force-velocity relationships. *Journal of Muscle Research and Cell Motility, 20*, 627–643.

d'Avella, A., & Bizzi, E. (2005). Shared and specific muscle synergies in natural motor behaviors. *Proceedings of the National Academy of Sciences USA, 102*, 3076–3081.

d'Avella, A., Saltiel, P., & Bizzi, E. (2003). Combinations of muscle synergies in the construction of a natural motor behavior. *Nature Neuroscience, 6*, 300–308.

Flash, T., & Hogan, N. (1985). The coordination of arm movements: An experimentally confirmed mathematical model. *Journal of Neuroscience, 5*, 1688–1703.

Harris, C. M., & Wolpert, D. M. (1998). Signal-dependent noise determines motor planning. *Nature, 394*, 780–784.

Hollerbach, J. M., & Flash, T. (1982). Dynamic interactions between limb segments during planar arm movement. *Biological Cybernetics, 44*, 67–77.

Jing, J., Cropper, E. C., Hurwitz, I., & Weiss, K. R. (2004). The construction of movement with behavior-specific and behavior-independent modules. *Journal of Neuroscience, 24*, 6315–6325.

Jordan, M. I., & Rosenbaum, D. A. (1989). Action. In M. I. Posner (Ed.), *Foundations of cognitive science*. Cambridge, MA: MIT Press.

Li, W., & Todorov, E. (2004). Iterative linear-quadratic regulator design for nonlinear biological movement systems. In *Proceedings of the First International Conference on Informatics in Control, Automation, and Robotics* (pp. 222–229).

Marr, D. (1982). *Vision*. New York: Freeman.

Mussa-Ivaldi, F. A., Giszter, S. F., & Bizzi, E. (1994). Linear combination of primitives in vertebrate motor control. *Proceedings of the National Academy of Sciences USA, 91*, 7534–7538.

Rosenbaum, D. A. (1991). *Human motor control*. San Diego: Academic Press.

Sanger, T. D. (1995). Optimal movement primitives. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), *Advances in neural information processing systems, 7*. Cambridge, MA: MIT Press.

Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In S. A. Solla, T. K. Leen, & K.-R. Müller (Eds.), *Advances in neural information processing systems, 12*. Cambridge, MA: MIT Press.

Thoroughman, K. A., & Shadmehr, R. (2000). Learning of action through adaptive combination of motor primitives. *Nature, 407*, 742–747.

Todorov, E. (2004). Optimality principles in sensorimotor control. *Nature Neuroscience, 7*, 907–915.

Todorov, E., & Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nature Neuroscience, 5*, 1226–1235.

Todorov, E., & Li, W. (2005). A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. *Proceedings of the 2005 American Control Conference* (Vol. 1, pp. 300–306).

Received August 16, 2005; accepted February 17, 2006.

Are People Successful at Learning Sequential Decisions on a Perceptual Matching Task?

Reiko Yakushijin (yaku@cl.aoyama.ac.jp)
Department of Psychology, Aoyama Gakuin University, Shibuya, Tokyo, 150-8366, Japan

Robert A. Jacobs (robbie@bcs.rochester.edu)
Department of Brain & Cognitive Sciences, University of Rochester, Rochester, NY 14627, USA

Abstract

Sequential decision-making tasks are commonplace in our everyday lives. We report the results of an experiment in which human subjects were trained to perform a perceptual matching task, an instance of a sequential decision-making task. We use two benchmarks to evaluate the quality of subjects' learning. One benchmark is based on optimal performance as defined by a dynamic programming procedure. The other is based on an adaptive computational agent that uses a reinforcement learning method known as Q-learning to learn to perform the task. Our analyses suggest that subjects learned to perform the perceptual matching task in a near-optimal manner at the end of training. Subjects were able to achieve near-optimal performance because they learned, at least partially, the causal structure underlying the task. Subjects' learning curves were broadly consistent with those of model-based reinforcement-learning agents that built and used internal models of how their actions influenced the external environment. We hypothesize that, in general, people will achieve near-optimal performances on sequential decision-making tasks when they can detect the effects of their actions on the environment, and when they can represent and reason about these effects using an internal mental model.

Keywords: sequential decision making; optimal performance; dynamic programming; reinforcement learning

Introduction

Tasks requiring people to make a sequence of decisions to reach a goal are commonplace in our lives. When playing chess, a person must choose a sequence of chess moves to capture an opponent's king. When driving to work, a person must choose a sequence of left and right turns to arrive at work in a timely manner. And when pursuing financial goals, a person must choose a sequence of saving and spending options to achieve a financial target. Interest in sequential decision-making tasks among cognitive scientists has increased dramatically in recent years (e.g., Busemeyer, 2002; Chhabra & Jacobs, 2006; Fu & Anderson, 2006; Gibson, Fichman, & Plaut, 1997; Gureckis & Love, 2009; Lee, 2006; Sutton & Barto, 1998; Shanks, Tunney, & McCarthy, 2002).

Here, we are interested in whether people are successful at learning to perform sequential decision-making tasks. There are at least two ways in which the quality of learning can be evaluated. These ways differ in terms of the benchmark to which the performances of a learner are compared. One way uses a benchmark of optimal performance on a task. Analyses based on optimal performance are referred to as ideal observer analyses, ideal actor analyses, or rational analyses in the literatures on perception, motor control, and cognition, respectively. At each moment during training with a task, a learner's performance can be compared to the optimal performance for that task. If a learner achieves near-optimal performance at the end of training, then it can be claimed that the learner has been successful.

A second way of evaluating a learner is to compare the learner's performances with those of an adaptive computational agent that is trained to perform the same task. We consider here an agent that learns via "reinforcement learning" methods developed by researchers interested in artificial intelligence (Sutton & Barto, 1998). Cognitive scientists have begun to use reinforcement learning methods to develop new theories of biological learning (e.g., Busemeyer & Pleskac, 2009; Daw & Touretzky, 2002; Schultz, Dayan, & Montague, 1997; Fu & Anderson, 2006). To date, however, there are few comparisons of the learning curves of people and agents based on reinforcement learning methods. Because reinforcement learning is regarded as effective and well-understood from an engineering perspective, and as plausible from psychological and neurophysiological perspectives, the performances of agents based on this form of learning can provide useful benchmarks for evaluating a person's learning. If a person's performance during training improves at the same rate as that of a reinforcement-learning agent, then it can be argued that the person is a successful learner. If a person's performance improves at a slower rate, then the person is not learning as much from experience as he or she could learn. Experimentation is often required to identify the cognitive "bottlenecks" preventing the person from learning faster. Lastly, if a person's performance improves at a faster rate, then this suggests that the person is using information sources or information processing operations that are not available to the agent. A new, more complex agent should be considered in this case.

We report the results of an experiment in which human subjects were trained to perform a perceptual matching task. This task was designed to contain a number of desirable features. Importantly, the perceptual matching task is an instance of a sequential decision-making task. Subjects made a sequence of decisions (or, equivalently, took a sequence of actions) to modify an environmental state to a goal state. In addition, efficient performance on the perceptual matching task required knowledge of how different properties of an environment interacted with each other. In many everyday tasks, people are required to understand the interactions, or "causal relations", among multiple components (Busemeyer, 2002; Gopnik &

Shulz, 2007). For example, when reaching for a coffee mug, a person must understand that forces exerted at the shoulder also influence the positions and velocities of the elbow, wrist, and fingers. To make an efficient movement, a person must use this knowledge of the causal interactions among motor components to design an effective motor plan.

Subjects' performances on the perceptual matching task were evaluated via two benchmarks. Using an optimization technique known as dynamic programming, optimal performance on this task was calculated. In addition, computer simulations of an adaptive agent were conducted in which the agent was trained to perform the perceptual matching task using a reinforcement learning method known as Q-learning (Sutton & Barto, 1998; Watkins, 1989). Comparisons of subjects' performances during training with optimal performance and with those of the adaptive agent suggest that: (i) subjects learned to perform the perceptual matching task in a near-optimal manner at the end of training; (ii) subjects learned, at least partially, the causal structure underlying the task; (iii) subjects' learning curves were consistent with those of model-based reinforcement-learning agents; and (iv) subjects may have learned by building and using mental models of how their actions influenced the external environment. Additional details and results are reported in Yakushijin & Jacobs (2010).

Experiment

Methods: Twenty-four undergraduate students at the University of Rochester participated in the experiment. Subjects were paid $10 for their participation. All subjects had normal or corrected-to-normal vision. Subjects were randomly assigned to one of six experimental conditions. Each condition included both training and test trials. Only the results of training trials are discussed here due to space limitations.

On a training trial, subjects performed a perceptual matching task which used visual objects from a class of parameterized objects known as "supershapes" (highly realistic but unfamiliar shapes; see Gielis, 2003). The parameters were latent (hidden) variables whose values determined the shapes of the objects. On each trial, subjects viewed a target object, a comparison object, and a set of six buttons (see left panel of Figure 1). Buttons were organized into three pairs, and each pair could be used to decrease or increase the value of an action variable. By pressing the buttons, subjects could change the values of the action variables which, in turn, changed the values of the parameters underlying the comparison object's shape which, in turn, changed the shape of the comparison object. Subjects' task was to press one or more buttons (i.e., to change the values of the action variables) to modify the shape of the comparison object until it matched the shape of the target object using as few button presses as possible.

An experimental condition was characterized by a specific set of causal relations among the latent shape parameters. For example, one such set is schematically illustrated in the right panel of Figure 1. Here, the three action variables are denoted A, B, and C. These variables are observable in the sense that subjects could directly and easily control their values through the use of the buttons. The values of the action variables determined the values of the shape parameters, denoted X, Y, and Z. Note that there are causal relations among the shape parameters. According to the network in Figure 1, if the value of X is changed, then this leads to a modification of Y which, in turn, leads to a modification of Z. The shape parameters determine the shape of the comparison object, whose perceptual features are denoted f_1, f_2, f_3, f_4, f_5, and f_6. The perceptual features used by a subject to assess the similarity of target and comparison object shapes may only be implicitly known by a subject, and may differ between subjects.

Importantly, to efficiently convert the comparison object's shape to the target object's shape (i.e., with the fewest number of button presses) often requires an understanding of the causal relations among the shape parameters. For instance, if the values of parameters X, Y, and Z all need to be modified, a person who does not understand the causal relations among shape parameters may decide to change the value of action variable C (thereby changing shape parameter Z), then the value of action variable B (thereby changing Y and Z), and finally the value of action variable A (thereby changing X, Y, and Z). In many cases, this will be an inefficient strategy. A person with good knowledge of the causal relations among the shape parameters knows that he or she can change the values of X, Y, and Z with a single button press that decreases or increases the value of action variable A. Thus, a good understanding of the causal relations among the shape parameters will lead to efficient task performance, whereas a poor understanding of the causal relations will lead to many more button presses than necessary.

The six experimental conditions differed in the causal relations among the latent shape parameters X, Y, and Z. Two of the causal relations were "linear" structures (one parameter had a direct causal influence on a second parameter which, in turn, had a direct causal influence on a third parameter; e.g., $X \rightarrow Y \rightarrow Z$ or $Y \rightarrow X \rightarrow Z$), two of the relations were "common cause" structures (one parameter had direct causal influences on the two remaining parameters; e.g., $Y \leftarrow X \rightarrow Z$ or $X \leftarrow Y \rightarrow Z$), and two of the relations were "common effect" structures (two parameters had direct causal influences on a third parameter; e.g., $X \rightarrow Y \leftarrow Z$ or $Y \rightarrow X \leftarrow Z$).

An experimental session consisted of 7 blocks of trials where a block contained a set of training trials followed by a set of test trials. (Test trials evaluated subjects' one-step look-ahead knowledge; on a test trial, a subject decided if a comparison object could be converted to a target object using a single button press, and the subject did not receive feedback. Again, test trials are not discussed here.) Each set contained 26 trials, one trial for each possible perturbation of a target object shape to form an initial comparison object shape.

Results: Task Performances: As a benchmark for evaluating subjects' performances on training trials, we computed optimal performances on these trials using an optimization method known as dynamic programming (Bellman, 1957).

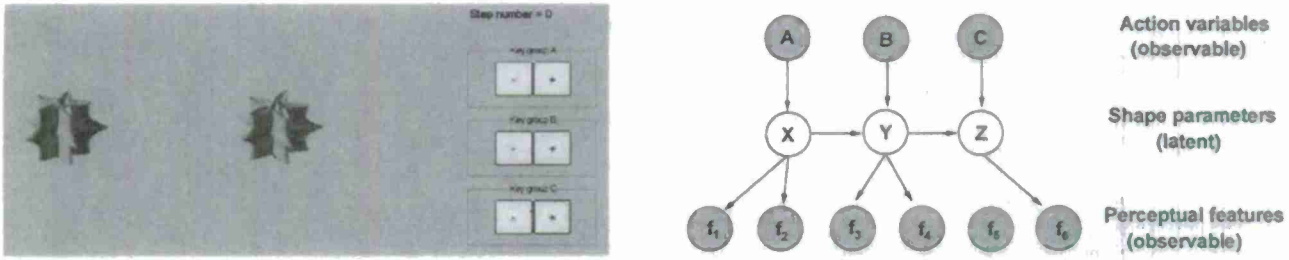

Figure 1: Left: Example of an experimental display. Right: Bayesian network representing the causal relations (in one of the experimental conditions) among the action variables, shape parameters, and perceptual features. For simplicity, the network does not represent the fact that subjects' button presses determined the values of the action variables.

In brief, dynamic programming is a technique for computing optimal solutions to multi-stage decision tasks. That is, dynamic programming finds the shortest sequences of actions that move a system from an initial state to a goal state when all states are fully observable. In the context of a training trial, the initial state corresponds to the initial values of the shape parameters X, Y, and Z for the comparison object, and the goal state corresponds to the values of the shape parameters for the target object. The dynamic programming algorithm is provided with full state information. This means that the algorithm knows the values of the comparison object's shape parameters at every time step. It also knows the state transition dynamics, meaning that it knows the causal relations among the shape parameters and, thus, knows how any button press will change the values of the shape parameters. Relative to our subjects, the dynamic programming algorithm is at an advantage. At the start of the experiment, our subjects did not know the values of the shape parameters or the causal relations among the parameters. Consequently, it would be impressive if subjects learned to perform the task as well as the dynamic programming algorithm.

We determined the optimal performances in the six experimental conditions via dynamic programming. Our analysis revealed that the range (1-5 steps or button presses) and the average length (2.54 steps) of the optimal action sequences were identical for all conditions. Thus, the conditions were well balanced in terms of their intrinsic difficulties.

Figure 2 shows subjects' learning curves on training trials in the two experimental conditions with linear causal structures among shape parameters. Due to space limitations, we do not show results for conditions with common-cause and common-effect structures, though subjects in these conditions showed very similar results to subjects in linear structure conditions (Yakushijin & Jacobs, 2010). Eight subjects participated in linear structure conditions and, thus, the figure contains eight graphs. The horizontal axis of each graph gives the block number, and the vertical axis gives the average difference between the number of steps (i.e., button presses) used by a subject during a trial and the optimal number of steps for that trial as computed by the dynamic programming procedure. These graphs show a number of interesting features. Many subjects found the task to be difficult toward the start of the experiment and, thus, their performances were highly sub-optimal during this time period. However, every subject learned during the course of the experiment. Importantly, every subject achieved near-optimal performance at the end of training: The average difference between a subject's performance and the optimal performance at the end of training is less than 1/2 of a step (mean = 0.434; standard deviation = 0.324).

Results: Causal Learning: The data from the training trials show that subjects achieved near-optimal performances. These results are consistent with the idea that subjects learned about the causal relations among the latent shape parameters. Additional analyses of training and test trials, not described here due to space limitations, confirm that subjects did indeed learn (at least partially) about these causal relations, and that this knowledge played a role in their task performances. Details can be found in Yakushijin & Jacobs (2010).

Reinforcement Learning Agents

Above, our analysis of subjects' data used a benchmark of optimal performance based on dynamic programming. Although very useful, this analysis does not allow us to evaluate the quality of subjects' rates of learning. To do so, we use a different benchmark based on an adaptive computational agent that uses a reinforcement learning method known as Q-learning to learn to perform the perceptual matching task (Sutton & Barto, 1998; Watkins, 1989). Without going into the mathematical details, the reader should note that Q-learning is an approximate dynamic programming method (Si et al., 2004). It is easy to show that, under mild conditions, the sequence of decisions found by an agent using Q-learning is guaranteed to converge to an optimal sequence found by dynamic programming (Watkins & Dayan, 1992). Hence, the benchmarks based on dynamic programming and on Q-learning are related.

In a reinforcement learning framework, it is assumed that an agent attempts to choose actions so as to receive the most reward possible. The agent explores its environment by assessing its current state and choosing an action. After executing this action, the agent will be in a new state, and will receive a reward (possibly zero) associated with this new state.

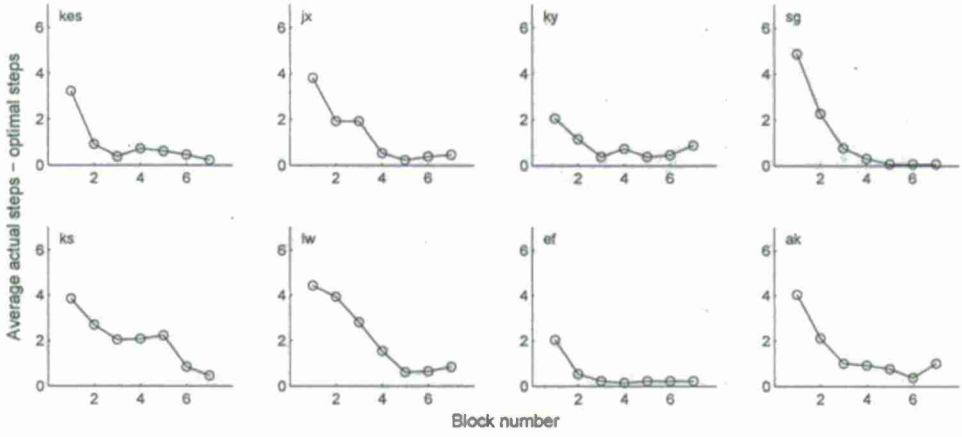

Figure 2: Subjects' learning performances on training trials in the two experimental conditions with linear causal structures among shape parameters (top row: $X \rightarrow Y \rightarrow Z$; bottom row: $Y \rightarrow X \rightarrow Z$).

The agent adapts its behavior in a trial-by-trial manner by noticing which actions tend to be followed by future rewards and which actions are not. To choose good actions, the agent needs to estimate the long-term reward values of selecting possible actions from possible states. Ideally, the value of selecting action a_t in state s_t at time t, denoted $Q(s_t, a_t)$, should equal the sum of rewards that the agent can expect to receive in the future if it takes action a_t in state s_t: $Q(s_t, a_t) = E[\sum_{k=0}^{\infty} \gamma_k r_{t+k+1}]$ where t is the current time step, k is an index over future time steps, r_{t+k+1} is the reward received at time $t+k+1$, and γ $(0 < \gamma \leq 1)$ is a term that serves to discount rewards that occur in the far future more than rewards that occur in the near future. An agent can learn accurate estimates of these ideal values on the basis of experience if it updates its estimates at each time step using the equation:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

where the agent makes action a_t in state s_t and receives reward r_{t+1}, and α is a step size or learning rate parameter (Sutton & Barto, 1998; Watkins, 1989).

In our first set of simulations in which a reinforcement-learning agent was trained to perform the perceptual matching task, all "Q-values" were initialized to zero, the discount rate γ was set to 0.7, and the learning rate α was set to 0.45. In preliminary simulations, these values were found to be best in the sense that they led to performances that most closely matched human performances. At each time step, the state of the agent represented the difference in shape between the comparison and target objects. It was a three-dimensional vector whose elements were set to the values of the shape parameters for the comparison object minus the values of these parameters for the target object. Six possible actions were available to the agent corresponding to the six buttons that a subject could press to modify the action variables. The agent chose an action using an ε-greedy strategy, meaning that the agent chose the action a that maximized $Q(s_t, a)$ with probability $1 - \varepsilon$ (ties were broken at random), and chose a random action with probability ε. The value of ε was initialized to one, and then it was slowly decreased during the course of a simulation. As a result, the agent tended to "explore" a wide range of actions toward the beginning of a simulation, and tended to "exploit" its current estimates of the best action to take toward the middle and end of a simulation. If the agent chose an action that caused the comparison object to have the same shape as the target object, the agent received a reward of 100. Otherwise, it received a reward of -1. The agent performed the training trials of the experiment in the same manner as our human subjects—it performed 7 blocks of training trials with 26 trials per block. To accurately estimate the agent's performances during training, the agent was simulated 1000 times.

The results for experimental conditions using linear causal structures are shown in the left graph of Figure 3 (results for other conditions were similar). The horizontal axis plots the block number, and the vertical axis plots the average difference between the number of steps (i.e., actions or button presses) used by the agent or by human subjects during a trial and the optimal number of steps for that trial as computed by the dynamic programming procedure (as in Figure 2; the error bars in Figure 3 indicate the standard deviations). The solid line shows the data for the simulated agent, and the dotted line shows the data for our human subjects. Interestingly, the learning curves of the simulated agent and of the human subjects have similar shapes, though subjects learned faster than the agent at nearly all stages of training in all experimental conditions. Modifications of the agent by either using different values for the agent's parameters or by adding "eligibility traces" did not significantly alter this basic finding.

Why did subjects show better learning performances than the simulated agent? In the machine learning literature, a distinction is made between model-free versus model-based reinforcement learning agents. The agent described above is an instance of a model-free agent. Although model-free agents are more common in the literature, we hypothesized

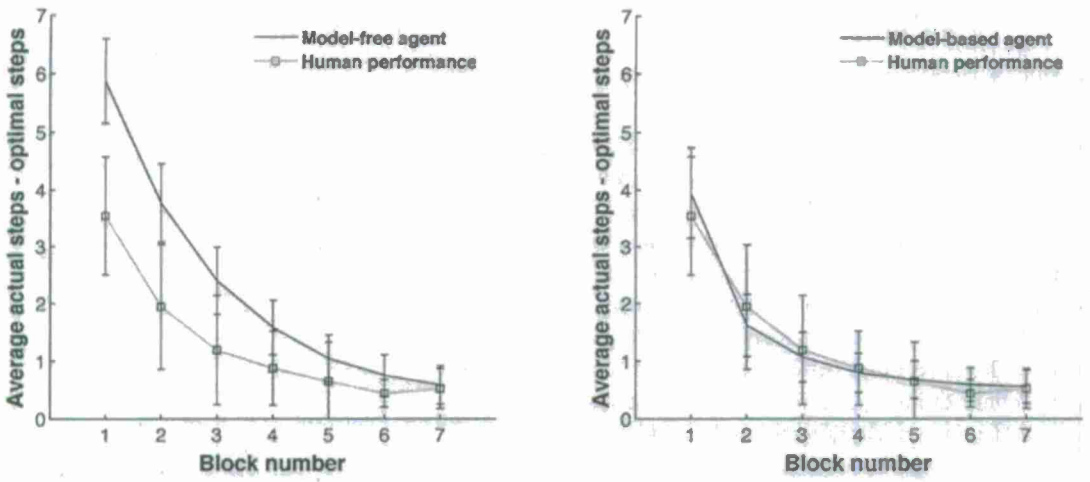

Figure 3: Left: Learning curves for the simulated agent trained via Q-learning (solid line) and for the human subjects (dotted line) in experimental conditions using linear causal structures (error bars plot standard deviations). Right: Identical to the left graph except that the simulated agent learned a model of how actions influenced the environment, and used this model to reason about good actions to take at each time step.

that a model-based reinforcement learning agent may provide a better account of our subjects' performances. Model-based agents typically learn faster than model-free agents, albeit with greater computational expense. Based on real-world experiences, a model-based agent learns an internal model of how its actions influence the environment. The agent updates its Q-values from both real-world experiences with the environment and from simulated experiences with the model (see Sutton and Barto, 1998, for details).

In our simulations, the model was an artificial neural network. Its six input units corresponded to the six possible actions or key presses (an action variable could either increase or decrease in value, and there were three action variables). Its nine output units corresponded to the nine possible influences on the comparison objects' shape parameters (a shape parameter could either increase in value, decrease in value, or maintain the same value, and there were three shape parameters). The network did not contain any hidden units.

When updating its Q-values, the model-based agent used 'prioritized sweeping' (Moore & Atkeson, 1993). This is an efficient method for focusing Q-value updates to state-action pairs associated with large changes in expected reward. Large changes occur, for example, when the current state is a non-goal state and the agent discovers a previously unfamiliar action that leads to a goal state. Large changes also occur when the current state is a non-goal state, and the agent discovers a new action that leads to a new non-goal state known to lie on a path toward a goal state.

In brief, our simulations used prioritized sweeping as follows. At each moment in time, the model-based agent maintained a queue of state-action pairs whose Q-values would change based on either real or simulated experiences. For each update based on a real experience, there were up to N updates based on simulated experiences. The items on the queue were prioritized by the absolute amount that their Q-values would be modified. For example, suppose that at some moment in time, state-action pair (s^*, a^*) had the highest priority. Then $Q(s^*, a^*)$ would be updated. If performing this update on the basis of simulated experience, the agent used the model to predict the resulting new state. In addition, the agent also used the model to examine changes to the Q-values for all state-action pairs predicted to lead to state s^*, known as predecessor state-action pairs. These predecessor state-action pairs were added to the queue, along with their corresponding priorities.

The simulations with the model-based agent were identical to those with the model-free agent. However, the model-based agent used different parameter values. Its discount rate γ was set to 0.3, its learning rate α was set to 0.05, and N, the number of Q-value updates based on simulated experiences for each update based on a real experience, was set to 5. In preliminary simulations, these values were found to be best in the sense that they led to performances that most closely matched human performances.

The combined results for the experimental conditions using linear causal structures are shown in the right graph of Figure 3 (once again, results for the other experimental conditions were similar). The learning curves of the model-based agent are more similar to those of human subjects than the curves of the model-free agent. Indeed, the curves of the model-based agent and of the human subjects are nearly identical. Our findings suggest (but do not prove) that subjects may have achieved near-optimal performances on the perceptual matching task by building internal models of how their actions influenced the external environment. By using these models to reason about possible action sequences, subjects quickly learned to perform the task.

Conclusions

Sequential decision-making tasks are commonplace in our everyday lives. Here, we studied whether people were successful at learning to perform a perceptual matching task, an instance of a sequential decision-making task. We used two benchmarks to evaluate the quality of subjects' learning. One benchmark was based on optimal performance as defined by a dynamic programming procedure. The other was based on an adaptive computational agent that used Q-learning to learn to perform the task. Overall, our analyses suggest that subjects learned to perform the perceptual matching task in a near-optimal manner. When doing so, subjects learned, at least partially, the causal structure underlying the task. In addition, subjects' learning curves were broadly consistent with those of model-based reinforcement-learning agents that built and used internal models of how their actions influenced the external environment.

The cognitive science literature now contains several studies of human performance on sequential decision-making tasks. Some studies have suggested that human performance is optimal, whereas other studies have suggested the opposite. To date, our field does not have a good understanding of the factors influencing whether people will achieve optimal performance on a task. Future research will need to focus on this critical issue. Previous articles in the literature suggested that perceptual aliasing (Stankiewicz et al., 2006) or the existence of actions leading to large rewards in the short-term but not the long-term (Neth, Sims, & Gray, 2006; Gureckis & Love, 2009) seem to be factors leading to sub-optimal performance. Here, we propose a new understanding of when people will (or will not) achieve optimal performance. We hypothesize that people will achieve near-optimal performance on sequential-decision making tasks when they can detect the effects of their actions on the environment, and when they can represent and reason about these effects using an internal mental model.

Acknowledgments

This work was supported by a Grant-in-Aid for Scientific Research (#20730480) from the Japan Society for the Promotion of Science, by a research grant from the Air Force Office of Scientific Research (FA9550-06-1-0492), and by a research grant from the National Science Foundation (DRL-0817250).

References

Bellman, R. (1957). *Dynamic Programming*. Princeton, NJ: Princeton University Press.

Busemeyer, J. R. (2002). Dynamic decision making. In N. J. Smelser & P. B. Baltes (Eds.), *International Encyclopedia of the Social and Behavioral Sciences*. Oxford, UK: Elsevier Press.

Busemeyer, J. R. & Pleskac, T. J. (2009). Theoretical tools for understanding and aiding dynamic decision making. *Journal of Mathematical Psychology, 53*, 126-138.

Chhabra, M. & Jacobs, R. A. (2006). Near-optimal human adaptive control across different noise environments. *The Journal of Neuroscience, 26*, 10883-10887.

Daw, N. D. & Touretzky, D. S. (2002). Long-term reward prediction in TD models of the dopamine system. *Neural Computation, 14*, 2567-2583.

Fu, W.-T. & Anderson, J. R. (2006). From recurrent choice to skill learning: A reinforcement-learning model. *Journal of Experimental Psychology: General, 135*, 184-206.

Gibson, F. P., Fichman, M., & Plaut, D. C. (1997). Learning in dynamic decision tasks: Computational model and empirical evidence. *Organizational Behavior and Human Decision Processes, 71*, 1-35.

Gielis, J. (2003). A generic geometric transformation that unifies a wide range of natural and abstract shapes. *American Journal of Botany, 90*, 333-338.

Gopnik, A. & Shulz, L. (2007). *Causal Learning: Psychology, Philosophy, and Computation*. New York: Oxford University Press.

Gureckis, T. M. & Love, B. C. (2009). Short-term gains, long-term pains: How cues about state aid learning in dynamic environments. *Cognition, 113*, 293-313.

Lee, M. D. (2006). A hierarchical Bayesian model of human decision-making on an optimal stopping problem. *Cognitive Science, 30*, 1-26.

Moore, A. & Atkeson, C. (1993). Prioritized sweeping: Reinforcement learning with less data and less real time. *Machine Learning, 13*, 103-130.

Neth, H., Sims, C. R., & Gray, W. D. (2006). Melioration dominates maximization: Stable suboptimal performance despite global feedback. *Proceedings of the 28th Annual Meeting of the Cognitive Science Society*.

Schultz, W., Dayan, P., & Montague, P. R. (1997). A neural substrate of prediction and reward. *Science, 275*, 1593-1598.

Shanks, D. R., Tunney, R. J., & McCarthy, J. D. (2002). A re-examination of melioration and rational choice. *Journal of Behavioral Decision Making, 15*, 233-250.

Si, J., Barto, A. G., Powell, W. B., & Wunsch, D. (2004). *Handbook of Learning and Approximate Dynamic Programming*. Piscataway, NJ: Wiley-IEEE.

Stankiewicz, B. J., Legge, G. E., Mansfield, J. S., & Schlicht, E. J. (2006). Lost in virtual space: Studies in human and ideal spatial navigation. *Journal of Experimental Psychology: Human Perception and Performance, 32*, 688-704.

Sutton, R. S. & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

Watkins, C. J. C. H. (1989). Learning From Delayed Rewards. Unpublished doctoral dissertation. Cambridge, UK: Cambridge, University.

Watkins, C. J. C. H. & Dayan, P. (1992). Q-learning. *Machine Learning, 8*, 279-292.

Yakushijin, R. & Jacobs, R. A. (2010). Are people successful at learning sequential decisions on a perceptual matching task? Manuscript submitted for journal publication.